Exploring or reducing noise? A global optimization algorithm in the presence of noise*

Didier Rullière †, Alaeddine Faleh ‡,
Frédéric Planchet §, Wassim Youssef ¶

Abstract

We consider the problem of the global minimization of a function observed with noise. This problems occurs for example when the objective function is estimated through stochastic simulations. We propose an original method for iteratively partitioning the search domain when this area is a finite union of simplexes. On each subdomain of the partition, we compute an indicator measuring if the subdomain is likely or not to contain a global minimizer. Next areas to be explored are chosen in accordance with this indicator. Confidence sets for minimizers are given. Numerical applications show empirical convergence results, and illustrate the compromise to be made between the global exploration of the search domain and the focalization around potential minimizers of the problem.

1 Introduction

Let us consider the problem of finding a global minimum of an objective function in the presence of noise, and when the search domain is a simplex (or a finite union of simplexes), for example when we try to find percentages that are summing to one. Since all percentages are summing to one, any percentage can be deduced from the others. It follows that considering $d+1$ percentages, $d \in \mathbb{N}^*$, the set of all possible input parameters can be represented by a simplex Θ in dimension d. This problem occurs in the context of finance, when considering asset allocations. Finding the right allocation is usually done by finding percentages of investments leading to minimizing a risk indicator (which can also take into account return considerations). In engineering problems, the problem of finding percentages that are summing to one appears in the mixture experiment field [cf. 7]. The same problem can be considered if the search domain can be represented as a finite union of simplexes (e.g. global optimization on a hypercube with linear constraints). In mechanical engineering the presence of a noise on the output is discussed in [14]. As mentioned in [36], traditional response surface methods often assume that experiments have sources of error, i.e. noise. The considered error may come from physical experiments or from computer experiments, from the use of simulation tools.

One considers a function $f : \Theta \mapsto \mathbb{R}$, where $\Theta \subset \mathbb{R}^d$. We assume that f is continuous, bounded, but not necessarily differentiable. In the case where $f(\theta)$ cannot be computed analytically, it is deduced from stochastic simulations. When this objective function is estimated through simulations, it involves estimation errors. The simulation model thus gives realizations of a noisy (random) function F:

$$F(\theta) = f(\theta) + \epsilon(\theta), \quad \theta \in \Theta_{\text{obs}}$$

where $\Theta_{\text{obs}} = \{\theta_1, \dots, \theta_m\}$ is the finite set of explored points, where $f(\theta) \in \mathbb{R}$ is the deterministic objective function. The random variables $\{\epsilon(\theta)\}_{\theta \in \Theta}$ represent noise due to simulations. We

*This work was presented in part at the Noisy Kriging-based Optimization (NKO) workshop, Bern, November 2010. It has been partially funded by ANR Research project ANR-08-BLAN-0314-01, by a grant from ANRT with reference 177/2008, by MIRACCLE-GICC project, and Chaire BNP Paribas Cardif *Management de la modélisation*.

†Université de Lyon, Université Lyon 1, Laboratoire SAF, Ecole ISFA. UCBL, ISFA, 50 Avenue Tony Garnier, F-69007 Lyon, France. didier.rulliere@univ-lyon1.fr; Corresponding author.

‡Université de Lyon, Université Lyon 1, Laboratoire SAF, Ecole ISFA; Caisse des Dépôts et Consignations.

§Université de Lyon, Université Lyon 1, Laboratoire SAF, Ecole ISFA; Winter & Associés.

¶Winter & Associés.

assume $\mathrm{E}[\epsilon(\theta)] = 0$, $\mathrm{V}[\epsilon(\theta)] < \infty$. We also assume that distinct elements of $\{\epsilon(\theta)\}_{\theta\in\Theta}$ are mutually independent.

The objective function can only be observed with noise: realizations of $F(\theta)$ are the only observable quantities, for $\theta \in \Theta_{\text{obs}}$. It follows that observations at each point θ usually need to be repeated in order to estimate $f(\theta)$. In this framework we want to estimate both:

1. The unique minimal value of f,
$$m^* = \inf_{\theta\in\Theta} \mathrm{E}[F(\theta)] = \inf_{\theta\in\Theta} f(\theta)\,.$$
2. The set of all parameters leading f to be close to this solution m^*:
$$\mathcal{S}_x = \{\theta \in \Theta, \mathrm{E}[F(\theta)] \leqslant x\}\ ,$$
for any given $x \in \mathbb{R}$ in the neighborhood of m^*.

The proposed algorithm consists of choosing adequate observations points in order to estimate these quantities.

Literature Various optimization algorithms have been proposed in the literature. When we look for a local minimum, some methods are well known, like gradient descent methods, Newton-Raphson, Hooke and Jeeves algorithm, method from [27], or specific methods related to some particular shapes of f, e.g. convexity. In the presence of noise, certain stochastic algorithms aiming to determine local optima or roots can also be considered [cf. 21, 6, 38, 30].

Even without noise, the problem of finding a global minimum rather than a local minimum is difficult, and this field is more recent [cf. 16]. In some sense, we need to ensure that the search domain is sufficiently explored. Without noise, Lipschitzian optimization, Schubert algorithm [cf. 34], DIRECT algorithm [cf. 18] or Efficient Global Optimization (EGO) [19] can be used. Interval methods are also used for structural optimization [cf. 35]. A review of global optimization methods for engineering applications is available in [3].

Here, we consider the problem of the global optimization of a function observed with noise. Furthermore, the objective function is not necessarily convex or differentiable. Some methods like simulated annealing [cf. 1, 8], genetic algorithms [cf. 2, 26, 41] or Evolution Strategies can be used [cf. 12, with application to shape optimization]. Some methods are derived from branch-and-bound algorithms [28]. Finally, some methods adapted from the construction of response surfaces are developed, either using kriging [cf. 22, 23, 29, 32] or other interpolation techniques [cf. 31]. Some algorithms rely on bayesian settings and information theory [cf. 4, 39]. Detailed studies on global optimization can be found in [24, 15, 11, 13, 40]. Many presentations of the *Workshop on Noisy Kriging-based Optimization* (Bern, november 2010) give some methods or illustrations of the widespread question of global optimization with noise.

The problem is here distinct from the one where the noise relies on input parameters of the objective function (uncertainty propagation), for which other approaches may be useful [cf. 17].

Among global optimization algorithms, many techniques in the literature rely on the choice of one exploration point among a finite set of candidates. Moreover, the optimization itself sometimes relies on the construction of a predictor. The computation of this predictor at several candidate points can be time consuming. In this paper, we propose an algorithm which eases the construction of the candidate set, and where the optimization time can be easily reduced.

Problem The optimization procedure will provide observations of $F(\theta_i)$ for some points $\theta_i \in \Theta_{\text{obs}}$. When observations are repeated at each point, a sample of observations $\{F_j(\theta_i)\}_{j=1\ldots n(\theta_i)}$ is created at each point $\theta_i \in \Theta_{\text{obs}}$. From these repeated observations, one can build an estimator $\widehat{f}(\theta_i)$ of $f(\theta_i)$. One can also get the estimation error $\widehat{\sigma}^2_e(\theta_i)$, which is the estimated variance of $\widehat{f}(\theta_i)$:

$$\left\{\begin{array}{lcl} \widehat{f}(\theta_i) & = & \frac{1}{n(\theta_i)}\sum_{j=1}^{n(\theta_i)} F_j(\theta_i)\,, \\ \widehat{\sigma}^2(\theta_i) & = & \frac{1}{n(\theta_i)-1}\sum_{j=1}^{n(\theta_i)} (F_j(\theta_i) - \widehat{f}(\theta_i))^2\,, \\ \widehat{\sigma}^2_e(\theta_i) & = & \frac{1}{n(\theta_i)}\widehat{\sigma}^2(\theta_i)\,. \end{array}\right. \tag{1}$$

Denote by m the number of elements in $\Theta_{\text{obs}} = \{\theta_1, \ldots, \theta_m\}$. The total number of observations is equal to $n(\theta_1)+\ldots+n(\theta_m)$. This raises the problem of the choice of both the set Θ_{obs} and the number of replicates required to estimate $\widehat{f}(\theta_i)$, $\theta_i \in \Theta_{\text{obs}}$.

In the financial field, the evaluation of a risk indicator may involve simulating multiple paths of some complicated random processes which can be time-consuming. In the engineering field, physical experiments as well as computer experiments may be expensive. Since computers are acting on finite sets in finite time, the evaluation budget of $F_j(\theta_i)$ is limited. Thus a compromise has obviously to be found between two alternatives:

- Exploring: Should one choose a large value of m and then small values of $n(\theta_i)$, $\theta_i \in \Theta_{\text{obs}}$? This would lead to noisy estimations $\widehat{f}(\theta_i)$ and a noisy optimal value.

- Or reducing noise: should one choose a little value for m and large values $n(\theta_i)$, $\theta_i \in \Theta_{\text{obs}}$? This would reduce the noise on $\widehat{f}(\theta_i)$, but with an insufficient exploration of Θ, and the proposed optimum might be a local one.

In the following, we consider a minimal value of n_0 replications of $F(\theta)$ for each exploration point θ (when necessary, the proposed algorithm will increase this quantity). Indeed, when the noise is non-homogeneous, the algorithm will rely on the estimation of σ_e^2, and we thus assume that $n_0 > 1$. Another way to estimate σ_e^2, allowing $n_0 = 1$ when using an assumption of a homogeneous noise, is to use kriging with an estimation of the noise effect (*nugget effect*) through empirical variograms or maximum likelihood estimation; the noise is then estimated globally, and does not necessarily rely on replications of $F(\theta)$ at same input points. This question is the one of the estimation of the (nonhomogeneous) noise amplitude, and the best way to estimate this noise may depend on the considered problem.

2 A global optimization algorithm

The proposed algorithm is based on a Branch-and-Bound algorithm. An iterative partition of the search domain will be constructed but, nonetheless, no area will be definitively excluded from the search. At each step the exploration of one area will be improved, and the choice of an area will depend on the probability that the area contains a minimizer, as will be explained in a further detailed model below.

In summary, denoting by $\mathfrak{Z} = \{Z_i\}_{i=1,2...}$ a partition of the initial search domain Θ, the basic principle of the algorithm consists in two steps (that will be detailed or slightly modified in following sections):

- A branching step: one area $Z^* \subset \Theta$ will be chosen and divided into two parts.

- An evaluation step: for each area Z, a quantity named *potential* will indicate if the area may contain a minimizer.

2.1 Partitioning the search domain

Consider an area Z of the search domain Θ. That is $Z \subset \Theta$ is a subset of the search domain. It seems quite convenient to choose a convex domain for Z, each point being then more easily linked with the explored vertices of the area.

There are many ways to separate a convex domain into several domains. This topic can be linked to triangulation topics [cf. Delaunay's triangulation 10]. Some optimization algorithms rely on a separation in several hypercubes of Θ (cf. DIRECT algorithm, [18]).

The branching step relies on some choices that are detailed here:

- The separation of an area $Z \subset \Theta$ will rely on the exploration of a set of n_Z new points of the area. In order to maximize the available information used for the choice of each point, we have chosen to restrict this set to only one point, that is $n_Z = 1$.

- Due to the initially considered shape of the search domain Θ which is a simplex of $\mathbb{R}^d$, we have here chosen to separate Θ into simplexes (and not into hypercubes or other convex areas).

- If an area Z is a simplex of $\mathbb{R}^d$, one can separate Z into $d+1$ simplexes by exploring a new point into the interior of Z. We decided here to place the new explored point in one edge of the simplex, thus dividing the simplex into two parts.

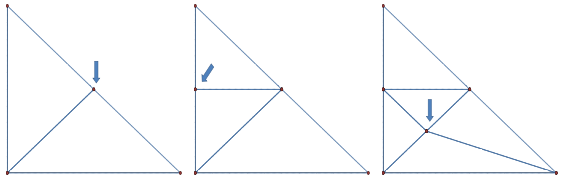

Figure 1: First iterations of a partition of a 2-simplex. Arrows indicate the added observed point at each step.

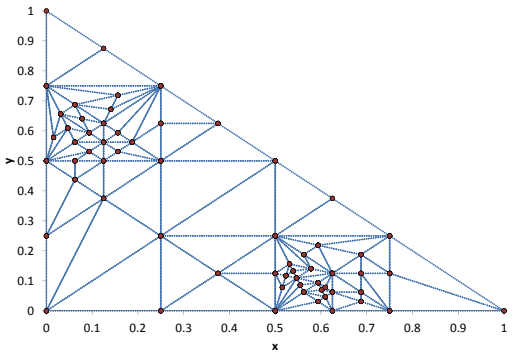

Figure 2: Partition of a 2-simplex into 60 areas. Here the search area is $\Theta = \left\{(x,y) \in [0,1]^2,\ x \leqslant y\right\}$.

- At last, the separating point will be chosen in the middle of one of the longer edges of the simplex: this will lead to a straightforward volume calculation of the two divided areas and avoids the creation of some very flat simplexes after successive separations.

The detailed dividing procedure of an area $Z \subset \Theta$ is given here. Consider an area $Z \subset \Theta$, simplex of Θ. Denote by $\{\theta_1, \dots \theta_{d+1}\}$ the $d+1$ vertices of this simplex Z. If one decides to divide Z into two areas, then Z_1 will be constituted of two simplexes Z_1 and Z_2, built as follows:

- Choose one of the longer edges $[\theta_{i_0}, \theta_{j_0}]$ of the simplex Z, $i_0, j_0 \in \{1, \dots, d+1\}$, $i_0 \neq j_0$.
- Define the separating point θ^+ as the middle of this longer edge, $\theta^+ = \frac{1}{2}(\theta_{i_0} + \theta_{j_0})$.
- Define the simplex Z_1 by its vertices:

$$(\{\theta_1, \dots \theta_{d+1}\} \setminus \theta_{i_0}) \cup \theta^+ \,,$$

  define by the same way Z_2 by its vertices:

$$(\{\theta_1, \dots \theta_{d+1}\} \setminus \theta_{j_0}) \cup \theta^+ \,.$$

It seems to us that this branching procedure would be very easy to implement, suited to the simplex. The exploration of only one point at each step, thus not depending on the dimension d of the problem is one of the advantages of such a branching step. At last, this separation methodology allows to explore, after successive steps, both frontiers and interiors of the considered areas.

In Figures 1 and 2, an example of an iterative partition of a simplex in dimension $d = 2$ is given. Note that, since at each step one point is explored on the edge of a simplex of the partition, it is possible without supplementary cost to divide adjacent areas, as shown in the last right illustration of the Figure 1.

Many optimization algorithms rely on the construction of a response surface, which allows to choose a point to explore among a set of candidates [cf. 20]. Computers are acting on finite sets in finite time, and this candidates set is necessarily finite. The segmentation of the search domain avoids the construction of this candidates set. This question is important since this candidates set should intuitively increase in size with the required precision on the minimizer.

2.2 Potential of each area

The idea of defining a potential is to see if an area is likely to contain a point leading to a lower minimum than the one which has been so far estimated. Consider that we have already explored a set of points $\Theta_{\text{obs}} = \{\theta_1, \dots, \theta_m\}$. For each point $\theta \in \Theta_{\text{obs}}$, we can calculate the estimator $\widehat{f}(\theta)$ and its variance, the estimation error, $\widehat{\sigma}_e^2(\theta)$ (cf. Equation 1), so that we can define a synthetic set of observations

$$T = \left\{ \left(\theta, \widehat{f}(\theta), \widehat{\sigma}_e^2(\theta)\right),\ \theta \in \Theta_{\text{obs}} \right\} . \qquad (2)$$

A very common way to build a predictor from these observations is to use a kriging predictor of f. In this case, one can define a random predictor $\tilde{f}_T(\theta)$

as a gaussian random variable. This variable is assumed to be centered on the kriging mean at point θ with a variance given by the kriging variance. Expressions of the kriging mean and variance as parametric functions of the set T and point θ are detailed for example in [22]. We just remark that the kriging predictor shall be adapted to the fact that observations are noisy. This leads to using kriging with adding a noise effect on the covariance matrix, corresponding to estimations errors (this noise effect is sometimes called *nugget effect*). Some details on possible kriging predictors are given in the numerical applications section. The kriging variance will depend on both a spatial error, due to the distance between the considered point and explored points, and an estimation error, due to estimation errors at explored points.

Suppose that we want to find the set of points leading the predictor $\tilde{f}_T$ of the function f to belong to a given interval J. For a minimization problem, one can consider for example $J =]-\infty, \widehat{m}^*]$, where $\widehat{m}^*$ is an estimation of the minimal value of f.

We first define a notion of the potential of an area Z, $Z \subset \Theta$.

Theorem 2.1 (Potential of an area)**.** *Given observations set T, and given a target interval J, the potential of an area Z is:*

$$\beta_{J,T}(Z) = V(Z) \cdot \mathrm{P}\left[\tilde{f}_T(\theta_Z) \in J\right] = \int_Z \mathrm{P}\left[\tilde{f}_T(\theta) \in J\right] d\theta$$

where $V(Z)$ is the d-volume of the area Z, and where θ_Z is a random point, with uniform distribution in Z.

For disjoint areas Z_1 and Z_2, it follows in particular that $\beta_{J,T}(Z_1 \cup Z_2) = \beta_{J,T}(Z_1) + \beta_{J,T}(Z_2)$.

Consider now a partition $\mathfrak{Z} = \{Z_1, \ldots, Z_n\}$ of the initial search domain Θ: all areas of $\mathfrak{Z}$ are disjoint, and their union is Θ. The area to be explored will be picked randomly, with a probability proportional to its potential. The following proposition gives more formally the probability to choose an area $Z \in \mathfrak{Z}$.

Theorem 2.1 (Probability to explore an area)**.** *Given a partition $\mathfrak{Z} = \{Z_1, \ldots, Z_n\}$ of Θ, assume that the probability $\rho_{J,T}(Z)$ to choose an area Z of $\mathfrak{Z}$ is proportional to its potential:*

$$\rho_{J,T}(Z) = \frac{\beta_{J,T}(Z)}{\sum_{Z_i \in \mathfrak{Z}} \beta_{J,T}(Z_i)}\,.$$

Then this probability is:

$$\rho_{J,T}(Z) = \mathrm{P}\left[\theta_U \in Z \middle| \tilde{f}_T(\theta_U) \in J\right],$$

where θ_U is a random point with uniform distribution on Θ.

Proof Since θ_Z has the same distribution as θ_U given that $\theta_U \in Z$, one can show that

$$\beta_{J,T}(Z) = V(Z)\mathrm{P}\left[\tilde{f}_T(\theta_U) \in J \middle| \theta_U \in Z\right].$$

Since $\mathrm{P}[\theta_U \in Z] = V(Z)/V(\Theta)$, by elementary conditional calculations one finds,

$$\beta_{J,T}(Z) = V(\Theta)\mathrm{P}\left[\tilde{f}_T(\theta_U) \in J \cap \theta_U \in Z\right].$$

Now write $\gamma_{J,T}(Z) = a\beta_{J,T}(Z)$, where a is a constant. Obviously,

$$\beta_{J,T}(Z) / \sum_{Z_i \in \mathfrak{Z}} \beta_{J,T}(Z_i) = \gamma_{J,T}(Z) / \sum_{Z_i \in \mathfrak{Z}} \gamma_{J,T}(Z_i)\,. \tag{3}$$

Assuming that $\mathrm{P}\left[\tilde{f}_T(\theta_U) \in J\right] > 0$, and setting $a^{-1} = V(\Theta)\mathrm{P}\left[\tilde{f}_T(\theta_U) \in J\right]$, we get

$$\gamma_{J,T}(\Theta) = 1\,. \tag{4}$$

Therefore

$$\gamma_{J,T}(Z) = \mathrm{P}\left[\theta_U \in Z \middle| \tilde{f}_T(\theta_U) \in J\right].$$

It follows that, for disjoint areas Z_1 and Z_2,

$$\gamma_{J,T}(Z_1 \cup Z_2) = \gamma_{J,T}(Z_1) + \gamma_{J,T}(Z_2)\,,$$

so that from Equation 4

$$\sum_{Z_i \in \mathfrak{Z}} \gamma_{J,T}(Z_i) = \gamma_{J,T}(\Theta) = 1\,, \tag{5}$$

and from Equations 3 and 5, we get

$$\rho_{J,T}(Z) = \gamma_{J,T}(Z) = \mathrm{P}\left[\theta_U \in Z \middle| \tilde{f}_T(\theta_U) \in J\right].$$

□

Proposition 2.1 is interesting since it gives a quite natural interpretation of the relative potential of each area: the probability to choose an area Z, given that a point θ_U chosen without preference

is interesting (i.e. $\tilde{f}_T(\theta_U) \in J$), is the probability that this point belongs to Z.

Consider a partition of the search domain in multiple small areas. From Proposition 2.1, one can see that the chosen heuristic is distinct from the maximization of an Expected Improvement [as defined in 19]: the potential is not an Expected Improvement, and we do not choose to explore directly the maximal potential area. This choice is relying on the idea that for identical (or roughly identical) potentials, the exploration should be uniform on the search domain: this leads to a sharing of the evaluation budget according to the potential of each area. This choice is easy to adapt to other shapes of target interval J, and to quantile inversion. One can also motivate this choice by the fact that, if the predictor $\tilde{f}_T$ does not change when T grows, the maximization of an Expected Improvement would lead to exploring always the same point. Exploring the area of the best potential thus seems to rely heavily on the goodness of fit of $\tilde{f}_T$. Some authors found that the search by the maximization of an Expected Improvement is too local when the estimated parameters are not good [33]. At last, it is possible to accentuate the differences between low and high potentials by simple distorsions (e.g. changing a potential $\beta \in [0,1]$ in β^γ, $\gamma > 1$): taking an extreme distortion (e.g. large γ) would lead to exploring only areas of maximal potential. As a summary, an area is supposed to be interesting if its potential is high. This occurs, on the one hand, if vertices of this area leads to estimations of the objective function close to the estimated minimum. On the other hand, an area may contain a minimizer if it is large enough, that is if it contains points far enough to vertices, that thus have to be explored. Intuitively, this last point relies heavily on the estimated regularity of the underlying objective function.

For practical evaluations of the potential $\beta_{J,T}(Z)$ when Z is a simplex, [37] shows how to sample a point uniformly on a simplex. In the case where one wishes to reduce the computation time of the potential (e.g. when evaluations of F are not so expensive, and many points have to be explored), the probability $\mathrm{P}\left[\tilde{f}_T(\theta_Z) \in J\right]$ may be replaced by a probability $\mathrm{P}\left[\tilde{f}_T(\theta_{B(Z)}) \in J\right]$ where $B(Z)$ is a point supposed to be representative of the area, e.g. its center. Even if this would change the considered probability, the relative potential of each area may not change a lot.

In [31], some remarks are done on links between potentials and Expected Improvement [as defined in 19]. An advantage of the potential is that its definition is suited to any target interval J. Considering other target interval than $]-\infty, m^*]$ may be interesting, for example, to find level curves of a function. In the insurance field or when analyzing risks, this is a common problem for solvency requirements. The constitution of a solvency capital in order to avoid ruin (or negative events) in a given percentage of scenarios, leads to finding quantiles of losses, and corresponding input parameters.

When one considers the minimization case, one can set for any area Z:

$$\beta_T(Z) = \beta_{J_T,T}(Z) \text{ with } J_T =]-\infty, \widehat{m}_T^*] \,, \quad (6)$$

where $\widehat{m}_T^*$ is an estimator of m^* given observations T. As an example, in further numerical illustrations we have chosen a conservative estimate of $\widehat{m}^*$ by using

$$\widehat{m}_T^* = \widehat{f}(\theta_T^*) + \lambda \widehat{\sigma}_e(\theta_T^*) \,, \quad (7)$$

where λ is a positive constant and where $\theta_T^* \in \Theta_{\text{obs}}$ is an actual explored minimizer. This is to avoid situations where, due to the estimation noise on m^*, the estimated value $\widehat{m}_T^*$ is lower than m^*, which would lead to considering the objective function of some interesting areas too far from this underestimated value $\widehat{m}_T^*$.
For the sake of clarity, the subscript T will be omitted when there is no ambiguity: the potential of an area Z will be simply denoted by $\beta(Z)$, and the estimator of the current minimum $\widehat{m}^*$.

2.3 Re-exploration of some points

In previous section, we have seen how to calculate the potential of each area belonging to a partition $\mathfrak{Z}$ of the search domain. This allows to choose at each step one area to be explored, for example by picking an area randomly, with a probability proportional to its potential.

Once an area is chosen for the exploration, one can wonder if it is necessary to partition this area

further and then create a subpartition $\mathfrak{Z}'$ of $\mathfrak{Z}$, or if it is more interesting to replicate observations of F at some vertices θ_i of the area, in order to reduce the noise relying on the estimated value of $f(\theta_i)$, $i \in \{1, \ldots, d+1\}$.

In this section, we will study this question. Should we divide an area around a point θ^+, middle of an edge $[\theta_1, \theta_2]$? or should we explore again θ_1 or θ_2? It is obvious that, in the absence of noise, re-exploring a vertex θ would lead to the same evaluation of $F(\theta)$ and is then useless. The re-exploration is justified only by the noise perturbing the evaluation of f.

Consider that an area Z has been chosen for the exploration. We have seen that the potential of an area was relying on the variance of the predictor (see Definition 2.1 on page 5, where the variance of $\tilde{f}_T$ is kriging variance). This potential is thus depending both on estimation errors and on spatial errors.

- Suppose first that we explore again a vertex of the area Z. since the area is not modified, the spatial structure will not change, but the estimation error at explored points will be reduced. This reduction is easy to estimate since the estimation error depends on the number $n(\theta)$ of repeated observations in θ. Write $\sigma^2(\theta) = \mathrm{V}[F(\theta)] = \mathrm{V}[\epsilon(\theta)]$ the variance of the random variable $F(\theta)$. The estimation error $\sigma_e(\theta)$ is the standard deviation of the empirical mean of a random sample of $n(\theta)$ observations of $F(\theta)$: $\sigma_e(\theta) = \sigma(\theta)/\sqrt{n(\theta)}$. Adding n^+ observations would lead to the new estimation error at point θ: $\sigma_e^+(\theta) = \sigma_e(\theta)\sqrt{n(\theta)}/\sqrt{n(\theta)+n^+}$, which is straightforward to estimate. We can thus estimate what would be the potential of the area after the exploration.

- Suppose now that we decide to separate the area Z into several parts around the separation point θ^+. It is quite easy to estimate $\sigma(\theta^+)$, and thus to estimate what would be the estimation error $\sigma_e(\theta^+)$ if one samples n^+ values of $F(\theta^+)$. The spatial structure on new separated areas is only depending on the known position of each vertex of Z and of θ^+ and, again, we can estimate the potential of each of the future separate part of Z.

In summary, before sampling some new values of $F(\theta)$, it is possible to estimate both the values of the potentials $\widehat{\beta}(Z_1)$ and $\widehat{\beta}(Z_2)$ in the case where we divide the area Z, and to estimate the potential $\widehat{\beta}(Z)$ of the area in the case where new replications of F are made on an already explored point.

The maximal potential over all areas constitute an indicator of the convergence of the algorithm:

$$\beta_{\max}(\mathfrak{Z}) = \max_{Z \in \mathfrak{Z}} \beta(Z)\,.$$

If this indicator is small for all areas, one can say that either the volume of the area is small (and the area have been explored), or the potential at the center of the area is small. Aiming at minimizing this maximal potential $\beta_{\max}(\mathfrak{Z})$, one can propose the following dividing rule:

$$\text{Divide if} \max\left\{\widehat{\beta}(Z_1), \widehat{\beta}(Z_2)\right\} \leqslant \widehat{\beta}(Z)\,. \quad (8)$$

This rule leads to a systematic division for non-noisy function F, systematic improvement of the estimation error on an already explored point if the estimation error is huge: this is a desired property since it is useless to reexplore the function at the same point when there is no noise.

Other dividing rules may be imagined, either based on other convergence criterions, or based on direct comparisons of measures of spatial errors and estimation errors.

2.4 Summary of the algorithm

The proposed algorithm is here summarized (cf. Algorithm 1). Given a partition of the search domain, the algorithm allows at each step to compute the potential of each subdomain of the partition. One then can pick randomly an area with a probability proportional to its potential. The chosen area is then explored. Depending on the choice to split this area or to reduce the noise at its vertices, this area is either divided into two parts, or replications of observations of the noisy function F are done at one vertex of the area.

Finally, when the algorithm has finished (e.g. when the evaluation budget of F is consumed), one can get for each subdomain of the partition the volume and the potential of the area. For each unexplored point of the search domain, one also can compute the potential of this point, without other

evaluations of F. This allows us to get the set of potential minimizers of the function f, given a confidence level s:

$$\widehat{\mathcal{S}}_{m^*,s} = \{\theta \in \Theta, \beta(\theta) \geqslant s\} \text{ with } \beta(\theta) = \mathrm{P}\left[\tilde{f}_T(\theta) \leqslant m^*_1\right]$$

3 Numerical illustrations

3.1 Kriging predictor and simplifications

In order to build a random predictor $\tilde{f}_T(\theta)$, we have chosen to interpolate estimated values of $\widehat{f}$ at some observed points, using observations set T (cf. Equation 2) and simple kriging. Kriging accounts for noise, which can be non-homogeneous, and is contained in the observation set T. This noise is here estimated through n_0 replications of $F(\theta)$, but n_0 can be set to 1 if the noise variance is given. We have chosen a Gaussian covariance matrix, where the covariance function $k(d)$ is supposed to depend only on the distance d between two distinct points:

$$k(d) = s^2 \exp(-(d/w)^2) \quad d > 0\,.$$

The question of the estimation of s and w is not studied here. We have chosen some parameters that seemed reasonable to us, without trying here to get the best possible estimations. The purpose was mainly to understand the behavior of the algorithm since fixing these parameters directly involve an arbitrage between local and global search. In section 3.2, we used $(s, w) = (10, 0.3)$. In section 3.3, we used either $(s, w) = (1, 0.3)$ (Figure 7, right), or $(s, w) = (0.1, 0.3)$ (other illustrations). These values are recalled when necessary.

We consider here the case where simulations that are done to determine one $F(\theta)$ are quite long, thus justifying a limited evaluation budget. Nevertheless, each evaluation of F does not necessarily take many hours, and the number of tested input parameters might thus be quite high (say several thousands). Many engineering situations also involve designs with several thousand parameters to test, with tests which may take several minutes each: we cannot always consider that all calculations of potentials are done in a negligible time compared to evaluations of $F(\theta)$, and this does not mean that testing all design points is fast. As an example, considering a predictor $\tilde{f}_T$ obtained by kriging is possible, if one assumes the existence of a non-stationary nugget effect linked with the non-homogeneous noise on f. It would however require, at each step, to invert matrices of size n_T^2, when n_T is the number of previously explored points, leading to heavy computations when n_T is large.
For figures presented in this section, we have chosen to reduce n_T in order to accelerate the construction of the kriging predictor: for calculating the potential of an area, only the $d+1$ vertices of this area were considered, thus involving matrices of size $(d+1) \times (d+1)$. This is a quite important reduction of the available information around a point, but this considerably speeds up all calculations. Depending on the considered problem, this reduction is not compulsory. One can imagine using more points for the kriging predictor, or using other interpolations techniques which do not imply the inversion of large matrices [cf. 31]. If simulations are very expensive, one may have time to build a full kriging predictor of the objective function.

In numerical illustrations of this section, we also took a slight modification of the potential of an area Z, defined as the product of the volume of the area and the potential of the center of the area. This will avoid considering multiple points of each area:

$$\bar{\beta}(Z) = V(Z) \cdot \mathrm{P}\left[\tilde{f}_T(\theta_{B_Z}) \in J\right],$$

Where θ_{B_Z} is the center of the area Z. This center will then be supposed to be representative of the area (rather than a uniform sample on the area). One should keep in mind that potentials are only used to compare relative weight of each area. In optimization procedures, interesting areas are supposed to become small, so that the uniform "coloration" of each area is not too problematic.

3.2 A basic illustration of the convergence behavior

In this first illustration section, we consider an example, as simple as possible, in order to investigate the empirical convergence of the algorithm in simplest cases. This raises the question of how to define the convergence.

Let us consider the set $\mathcal{S}$ of minimizers of f and the set B_r of all points at a given distance r from

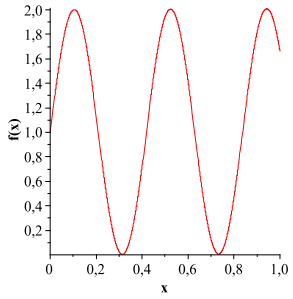

Figure 3: Objective function $f(x) = 1 + sin(15x) + \alpha x$, $\alpha = 0.01$, without noise.

a solution of $\mathcal{S}$.

$$B_r = \bigcup_{S \in \mathcal{S}} B_r(S) \text{ with } B_r(S) = \{\theta \in \Theta,\ d(\theta, S) \leqslant r\}$$

One can define the proportion of evaluations of F in B_r among all evaluations, which is defined as:

$$p_r = \frac{\sum_{\theta_i \in \Theta_{\text{obs}} \cap B_r} n(\theta_i)}{\sum_{\theta_i \in \Theta_{\text{obs}}} n(\theta_i)}\,. \tag{9}$$

where Θ_{obs} is the set of explored points. When $n(\theta_i) = n_0$ for any $\theta_i \in \Theta_{\text{obs}}$,

$$p_r = \text{card}(\Theta_{\text{obs}} \cap B_r)/\text{card}(\Theta_{\text{obs}})\,, \tag{10}$$

and p_r is also the proportion of explored points in B_r among all explored points (we recall that Θ_{obs} is the set of explored points).

If one considers a uniform exploration of Θ, it is obvious that, without any knowledge of f, any exploration has a probability $V(B_r)/V(\Theta)$ of being at a distance lower than r from a solution, $V(.)$ being the volume of an area in the dimension d of the problem. Given n exploration points uniformly distributed on Θ, the probability that one of these points is at a distance lower than r from a solution is thus given by a geometric distribution, and tends to 1 when n increases.

The minimal distance between the explored set and one minimizer is an interesting indicator of convergence, but alone is not sufficient to get an idea of the ability of one algorithm to perform better than a uniform exploration.

We study now the convergence of the algorithm in one of the most simple cases one can imagine, on a basic test function given in Figure 3, without noise and in dimension $d = 1$. We draw the empirical distribution of explored points, on the left side of Figures 4 and 5. In these figures, we can observe the distribution of explored points for different values of n, where n is the number of explored points. One empirically sees that when the size of the explored set increases, this distribution gets closer to a dirac distribution, or to a mixed-dirac distribution in the case of two minimizers (Figure 4). On the right part of Figures 4 and 5, we get an idea of the proportion p_r of points belonging to B_r among explored points. In both cases, this proportion increases, and gets closer to one when the number of explored points increases.

What is particularly noticeable here is that even when the test function have a second local optimum very close to the global one, as in Figure 3, the algorithm first considers this potential second solution, then rapidly focuses on the correct global solution as one can see in Figure 5. A uniform exploration of the search domain would here lead to a uniform distribution of explored points on the left of Figures 4 and 5. It would also lead to a roughly constant small proportion of explored points belonging to B_r on the right part of these figures, for a distance $r = 0.01$. In some sense, on this very simple example, the algorithm shows its ability to exploit the information of previously explored points and to perform better than a uniform exploration. It empirically converges towards an exploration around the global minimizers of the objective function.

3.3 An illustration with noise in dimension 2

We propose in this paragraph some illustrations with the following random function, defined on the orthogonal unit simplex Θ in dimension $d = 2$, for any point $\theta = (x, y)$ in Θ:

$$\begin{aligned} f(\theta) &= (\min(x, y) - 0.1)^2 + (\max(x, y) - 0.6)^2\,, \\ F(\theta) &= f(\theta) + \sigma_B(U - 0.5)\,, \end{aligned}$$

The variable U is a uniform random variable on $[0, 1]$. In each design point θ, $n_0 = 10$ simulations were made in order to estimate both $f(\theta)$ and $\sigma_e^2(\theta)$. The noise is here homogeneous, and thus estimations of the variance $\sigma^2(\theta)$ are very close for different sites θ. These variances were here

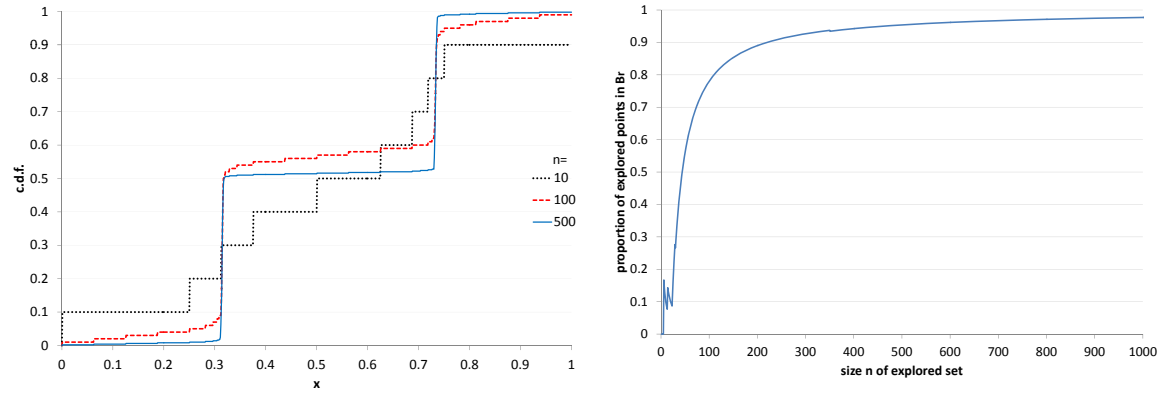

Figure 4: Cumulative distribution function (c.d.f.) of explored points for $n = 10$, $n = 100$ or $n = 500$ explored points (left) and proportion $p_r = \text{card}(\Theta_{\text{obs}} \cap B_r)/\text{card}(\Theta_{\text{obs}})$ of points at a given distance to a solution after $n = \text{card}(\Theta_{\text{obs}})$ explorations (right). Underlying objective function $f(x) = 1+sin(15x)+\delta x$ in the case $\delta = 0$ (two global minimizers).

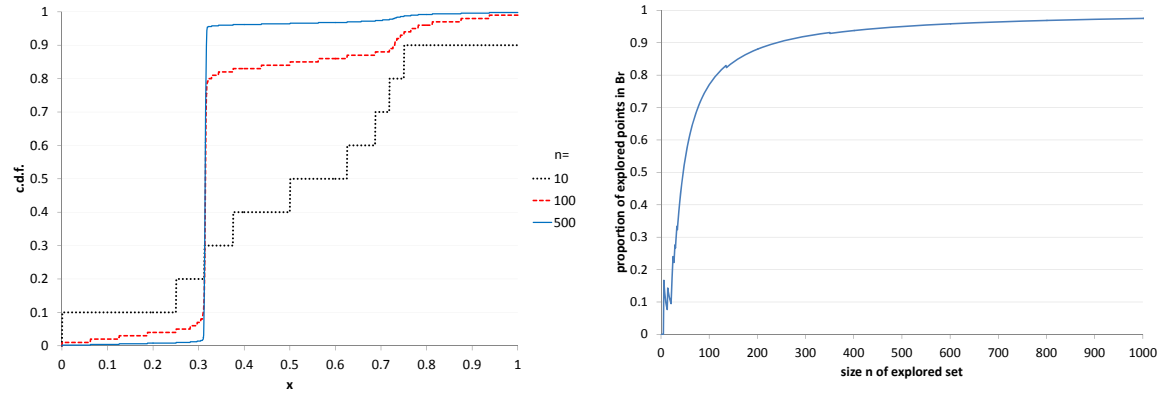

Figure 5: Cumulative distribution function (c.d.f.) of explored points for $n = 10$, $n = 100$ or $n = 500$ explored points (left) and proportion $p_r = \text{card}(\Theta_{\text{obs}} \cap B_r)/\text{card}(\Theta_{\text{obs}})$ of points at a given distance to a solution after $n = \text{card}(\Theta_{\text{obs}})$ explorations (right). Underlying objective function $f(x) = 1+sin(15x)+\delta x$ in the case $\delta = 0.01$ (one global minimizer).

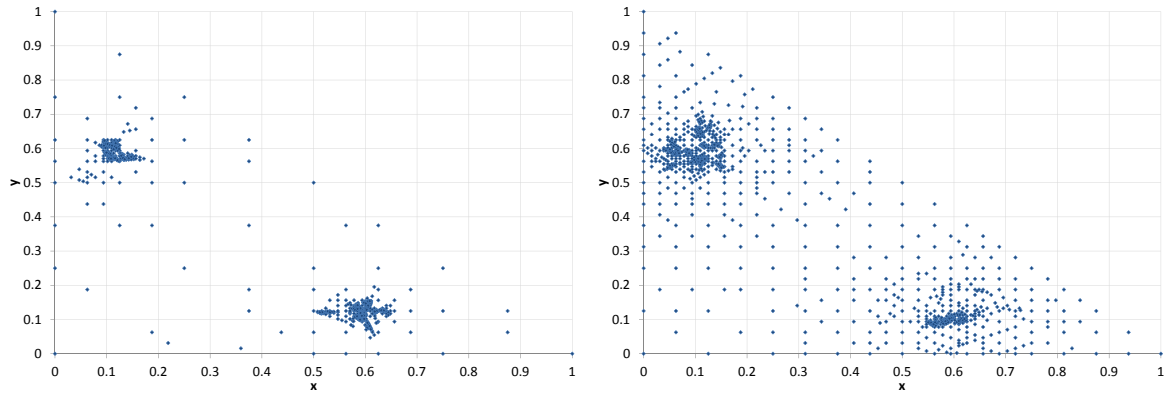

Figure 7: With noise or with overestimated kriging variance: set of simulation points F with noise $\sigma_B = 0.1$, $s = 0.1$ (left) or $\sigma_B = 0.1$, $s = 1$ (right), without re-exploration criterion. $w = 0.3$ and iterations number is $n = 1000$ in both cases.

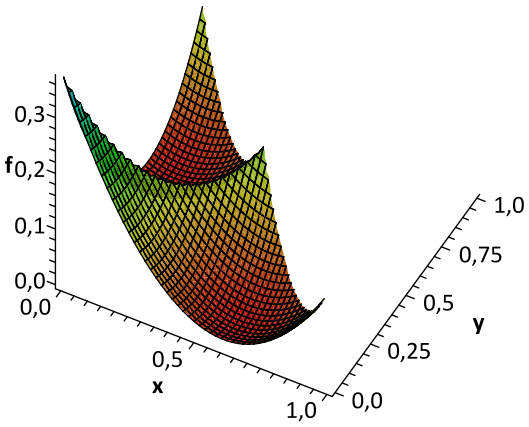

Figure 6: Shape of the function $f(\theta) = (\min(x, y) - 0.1)^2 + (\max(x, y) - 0.6)^2$, with $\theta = (x, y)$.

estimated as if this homogeneity information was lost. The reduction of n_0 is possible (even down to $n_0 = 1$), if the variance of the noise is given, or correctly estimated through the limit behavior of the empirical variogram at small distances or by other statistical techniques. The shape of the function f is given in the Figure 6.

We first consider the behavior of the algorithm without re-exploration, that is replacing the re-exploration criterion by a systematic division: each interesting area is systematically divided into two parts. In Figure 7, we can see that the algorithm explores the whole search domain, but focuses around minimizers. In practice it focuses more rapidly without noise and when the function is supposed to be regular enough. On the left part of this Figure 7, one can see what happens with a noise, some areas around the minimizers are heavily explored in order to ensure that they do not potentially contain a minimizer. On the right part of this Figure 7, we deliberately gave a higher value of s, leading to a higher kriging variance: the algorithm thus assumes that the objective function is less regular than it is, and explores more carefully the whole search domain. Choosing very high values of s leads to a roughly uniform exploration of the search domain. The covariance parameters can thus be seen as a way to privilege either the local or the global search.

In the Figure 8, we give the shape of the simulation points with the re-exploration criterion. Points

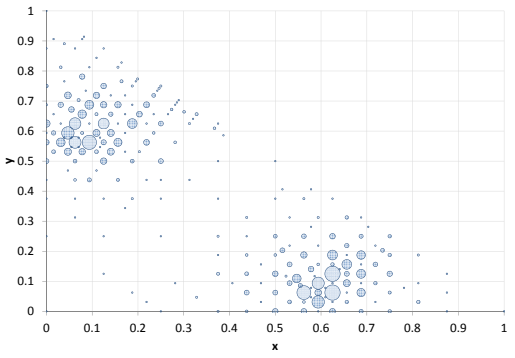

Figure 8: Simulation points for a noise $\sigma_B = 0.1$, with re-exploration criterion, $(s, w) = (0.1, 0.3)$. The number of iterations (exploration or re-exploration) is $n = 1000$. The size of bubbles at each explored point $\theta = (x, y)$ are proportional to the number $n(\theta)$ of evaluations of $F(\theta)$.

are re-explored and thus fewer sites are observed. The distance between the set of explored points to a real minimizer may thus be greater than the one with a systematic subdivision of the search domain. The re-exploration criterion may be useful if one wishes to ensure a better accuracy of the estimated value of f at observed points. It can also be important in engineering problems when each design point is expensive not only in time but also in cash, when one needs to reduce design points (e.g. when each design point requires a specific measurement device, for example a meteorology station). We observe that the algorithm avoids repeating observations far from minimizers, and reduces the noise more frequently when f is close to its minimal value.

3.4 Comparison with other algorithms

Let us use the same objective function as in previous Section 3.3. We present here a very short benchmark, to get an idea of the performance of the algorithm compared with other algorithms. We define first several indicators in order to check the global exploration of the objective function around each minimizer. For a point $\theta \in \Theta$ and the finite set Θ_{obs} of explored points, $\Theta_{\text{obs}} \subset \Theta$, define first the distance between θ and the set Θ_{obs} as $d(\theta, \Theta_{\text{obs}}) = \min_{\theta_{\text{obs}} \in \Theta_{\text{obs}}} d(\theta, \theta_{\text{obs}})$. Consider now the (here finite) solution set $\mathcal{S}$ of minimizers of f. This set is supposed to be known (which is usually the case for test functions only).
The best and worst explored solutions are defined respectively as

$$S^- = \arg\min_{S \in \mathcal{S}} d(S, \Theta_{\text{obs}}) \text{ and } S^+ = \arg\max_{S \in \mathcal{S}} d(S, \Theta_{\text{obs}}),$$

with corresponding distances to explored set

$$d^- = d(S^-, \Theta_{\text{obs}}) \text{ and } d^+ = d(S^+, \Theta_{\text{obs}}).$$

These distances aim at determining the ability of an algorithm to find a local or a global solution:

- The distance d^- is small if there exists a solution point close to the set of explored points (good local exploration of one solution).
- The distance d^+ is small if all solution points are close to the set of explored points (good global exploration of all solutions).

Recall $B_r(S) = \{\theta \in \Theta,\, d(\theta, S) \leqslant r\}$. The proportion of evaluations of F around a solution S is defined as

$$p_r(S) = \frac{\sum_{\theta_i \in \Theta_{\text{obs}} \cap B_r(S)} n(\theta_i)}{\sum_{\theta_i \in \Theta_{\text{obs}}} n(\theta_i)}.$$

The best and worst proportions are then defined respectively as

$$p_r^+ = \max_{S \in \mathcal{S}} p_r(S) \text{ and } p_r^- = \min_{S \in \mathcal{S}} p_r(S).$$

Consider a given global minimizer S of the objective function. The proportion $p_r(S)$ gives the relative number of evaluations of F in the neighborhood of this solution point S (among all evaluations of F). This proportion is high if many exploration points were at distance lower than r from the solution S. Now considering all known global minimizers:

- The best proportion p_r^+ is high if the neighborhood of at least one solution point has been explored. It thus gives an indication of the ability of the algorithm to find *one* solution point.
- The worst proportion p_r^- is high if the neighborhood of all solution points have been explored. It thus gives an indication of the ability of the algorithm to find *all* solution points.

In short, as previous indicators d^- and d^+, p_r^+ gives an indication of the local performance of the algorithm, whereas p_r^- gives an indication of the global performance of the algorithm. The indicators p_r^- and p_r^+ also indicate how many explorations were made in the neighborhood of each minimizer. With two global minimizers, the sum $p_r^- + p_r^+$ indicates the relative number of explored points in the whole neighborhood B_r of minimizers. When p_r^- is close to p_r^+, the algorithm uses similar exploration budget for each minimizer, which indicates a good equilibrium of the exploration. In the presence of two minimizers, best values of (p_r^-, p_r^+) are thus close to $(50\%, 50\%)$ if we look for all global minimizers, or $(0, 100\%)$ if we look for one minimizer only.

At last, the dispersion of the estimator $\widehat{f}$ in the neighborhood of a solution is measured by

$$\sigma_e(B_r) = \text{average}\left\{\widehat{\sigma}_e(\theta_i),\ \theta_i \in \Theta_{\text{obs}} \cap B_r\right\} .$$

This indicator is the only one which is using the values of F at explored points. It is particularly useful to see how much can the noise be reduced with re-exploration strategies.

The algorithm has a stochastic behavior. In order to quantify its behavior, we will also present some usual statistics of these performance measures over a given number of runs. Random numbers are issued from a Mersenne Twister generator, using in all cases the same initial seed, which was chosen before running the algorithms.

For comparison of performances with other algorithms, we tried on the same test function several algorithms, with the same evaluation budget of $n = 1000$ iterations (explorations or re-explorations):

- *Systematic Scission and Reexploration algorithms* are the two variants of the algorithm presented in this paper, using respectively a systematic scission procedure (scission criterion always set to the value *true*), or using the scission criterion of Equation 8.

- *EGO adaptation* is an adaptation of EGO algorithm [cf. 19]. At each step of the algorithm, explored points allow to built a kriging predictor of the function f. One uses this predictor to compute an Expected Improvement [19] for a set of candidate points. The candidate point with highest Expected Improvement is then chosen for exploration. Practically, we used 1000 candidate points at each step, chosen uniformly over the initial simplex [37]. Due to a too high complexity of this algorithm, the kriging predictor at each step was built using the last 500 explored points (instead of 3 points per area in our algorithm in dimension $d = 2$). As in our algorithm, kriging parameters are given at the beginning of the algorithm and are not re-estimated at each step, and refinements may be found to propose better suited candidate points, or for improving the kriging predictor. The computation of f is here very fast compared to the optimization itself, and thus this variant of EGO was more than several thousand times slower than our algorithm. For this reason, we only used 30 runs for this algorithm, instead of 1000 runs for other algorithms.

- *Kiefer-Wolfowitz algorithm (KWB)* We also tried the Kiefer-Wolfowitz-Blum algorithm [21, 6, 9], initialized with quite standard parameters (the one proposed in the first page of [21], $a_n = (n + \nu)^{-1}$ and $c_n = (n + \nu)^{-1/3}$, where the integer parameter ν has been added and chosen as the one giving the best average optimization results on a large number of runs, leading here to $\nu = 262$). On each run of this algorithm, the starting position was chosen randomly, uniformly on the search domain [cf. 37]. This algorithm is a stochastic optimization algorithm designed for finding one unique solution on a convex part of a noisy function. It is given here in order to compare the ability of another algorithm to find one local minimizer of the function.

- *Genetic Algorithm (GA)* We also tried a standard genetic algorithm. The considered algorithm is given in [25]. The parameters used are 50 generations of 20 individuals, with an elite of 5 individuals and a mutation probability of 20%. These parameters were not deeply optimized, but changes on the chosen parameters had little impact on the performance indicators.

The Table 1 gives the mean, standard deviation (*std-dev*), minimum and maximum (min and max), and some quantiles (q_{25}, q_{50}, q_{75}) of performance in-

with noise: $\sigma_B = 0.1$

algorithm	indicator	mean	std-dev	min	q_{25}	q_{50}	q_{75}	max
EGO	d^-	2.73E-03	1.95E-03	3.72E-04	**1.25E-03**	**2.10E-03**	3.94E-03	**7.94E-03**
(30 runs)	d^+	6.26E-03	**3.63E-03**	2.14E-03	3.20E-03	5.04E-03	9.74E-03	**1.64E-02**
KWB	d^-	4.52E-02	2.92E-02	3.25E-04	2.62E-02	4.73E-02	5.77E-02	2.56E-01
(1000 runs)	d^+	4.96E-01	1.25E-01	2.46E-01	3.95E-01	4.96E-01	6.22E-01	6.90E-01
GA	d^-	6.31E-03	4.05E-03	5.12E-05	3.23E-03	5.57E-03	8.56E-03	2.93E-02
(1000 runs)	d^+	1.69E-02	8.31E-03	9.99E-04	1.06E-02	1.56E-02	2.19E-02	6.13E-02
Systematic Scission	d^-	**2.15E-03**	**1.65E-03**	**1.38E-04**	1.42E-03	2.21E-03	**2.21E-03**	8.84E-03
(1000 runs)	d^+	**5.47E-03**	4.69E-03	**5.52E-04**	**2.21E-03**	**4.18E-03**	**8.84E-03**	3.54E-02
Reexploration	d^-	4.39E-03	3.83E-03	5.52E-04	2.21E-03	2.21E-03	8.21E-03	3.54E-02
(1000 runs)	d^+	1.02E-02	8.68E-03	1.42E-03	5.95E-03	8.84E-03	**8.84E-03**	3.54E-02

without noise: $\sigma_B = 0$

algorithm	indicator	mean	std-dev	min	q_{25}	q_{50}	q_{75}	max
EGO	d^-	1.10E-03	6.83E-04	1.24E-04	5.76E-04	9.33E-04	1.51E-03	3.10E-03
(30 runs)	d^+	4.08E-03	1.95E-03	1.15E-03	2.59E-03	3.69E-03	5.07E-03	9.39E-03
KWB	d^-	4.57E-02	2.96E-02	2.30E-04	2.70E-02	4.77E-02	5.79E-02	3.02E-01
(1000 runs)	d^+	4.97E-01	1.25E-01	2.46E-01	3.95E-01	4.96E-01	6.23E-01	6.90E-01
GA	d^-	1.12E-03	8.26E-04	8.13E-06	5.30E-04	9.25E-04	1.50E-03	6.74E-03
(1000 runs)	d^+	1.89E-02	8.89E-03	5.73E-04	1.22E-02	1.79E-02	2.44E-02	7.01E-02
Systematic Scission	d^-	**5.08E-06**	**3.37E-06**	**3.15E-07**	**1.80E-06**	**5.81E-06**	**8.63E-06**	**8.63E-06**
(1000 runs)	d^+	**8.07E-06**	**1.76E-06**	**3.63E-07**	**8.63E-06**	**8.63E-06**	**8.63E-06**	**1.10E-05**

Table 1: Usual statistics for performance measures d^- and d^+ over several runs. Case with noise, $\sigma_B = 0.1$ (up), or without noise $\sigma_B = 0$ (down). Common parameters are $s = 0.1, w = 0.3, \lambda = 2, n = 1000$. Best results are indicated in bold font.

noise	algorithm	runs	d^-	d^+	p_r^-	p_r^+	$\sigma_e(B_r)$	t_{run}
$\sigma_B = 0.1$	EGO	30	2.73E-03	6.26E-03	30%	35%	9.12E-03	2675
	KWB	1000	4.52E-02	4.96E-01	0%	39%	9.13E-03	$<$ **0.01**
	GA	1000	6.31E-03	1.69E-02	4%	24%	9.13E-03	< 0.03
	Systematic Scission	1000	**2.15E-03**	**5.47E-03**	**36%**	**57%**	9.13E-03	1
	Reexploration	1000	4.39E-03	1.02E-02	30%	56%	**6.92E-03**	0.7
$\sigma_B = 0$	EGO	30	1.10E-03	4.08E-03	1%	**95%**	$<$ 2E-11	2678
	KWB	1000	4.57E-02	4.97E-01	0%	39%	$<$ 2E-11	$<$ **0.01**
	GA	1000	1.12E-03	1.89E-02	3%	26%	$<$ 2E-11	< 0.03
	Systematic Scission	1000	**5.08E-06**	**8.07E-06**	**40%**	51%	$<$ 2E-11	1.2

Table 2: Average values of indicators over several runs. Case with noise, $\sigma_B = 0.1$ (up), or without noise $\sigma_B = 0$ (down). Common parameters are $s = 0.1, w = 0.3, \lambda = 2, r = 0.01$. The column t_{run} gives an indication of relative execution times per run, with base 1 for Systematic Scission algorithm when $\sigma_B = 0.1$. Best results are indicated in bold font.

dicators d^- and d^+ over a given number of runs. The Table 2 gives the average values of d^-, d^+, p_r^-, p_r^+ and $\sigma_e(B_r)$ over all runs.

With noise, the small number of runs of the *EGO* variant algorithm does not allow to conclude if this algorithm is more efficient than others or not, but it gives an idea of its performance. One can check here that performance indicators of this *EGO* variant in the presence of noise are of the same order as the *Systematic Scission* algorithm (see Tables 1 and 2). The *Reexploration* algorithm leads to higher average distances d^- and d^+ than the *Systematic Scission* algorithm since less points are explored (see Table 1). However, it leads to a better knowledge of the objective function around the minimizers: the average noise $\sigma_e(B_r)$ is reduced compared to the *Systematic Scission* algorithm (see Table 2). In both cases, with *Systematic Scission* or with *Reexploration*, one can see that the average proportions of explored points around the best and the worst solutions (p_r^- and p_r^+) are quite well equilibrated.

Without noise, *Systematic Scission* and *Reexploration* algorithms lead to the same results, since no reexploration is done when $\sigma_B = 0$ (which is a desired property). These two algorithms behave very well in this case. One major interest of the scission procedure is that it leads to an efficient choice of candidate points: *EGO* variant is here penalized by the selection of the best point among only 1000 candidates, whereas the partition of the search domain in the *Systematic Scission* algorithm leads to a choice among candidates mainly located around the minimizers. This kind of dichotomic selection allows a very good accuracy for the location of the minimizers (see Tables 1 and 2).

The Kiefer-Wolfowitz-Blum (KWB) algorithm is not a global optimization algorithm, and it illustrates here the difference of approach with other global optimization algorithms: the exploration of the search domain is poor. This stochastic algorithm is thus to be avoided if its application conditions are not fulfilled, and in particular if the supposed convexity of the objective function does not hold. An illustration of the behavior of KWB algorithm is given in Figure 10. We can see in Tables 1 and 2 that the Kiefer-Wolfowitz-Blum algorithm performs less efficiently than the other algorithms, even for finding only one of the two minimizers (i.e. even when considering only the best distance d^-).

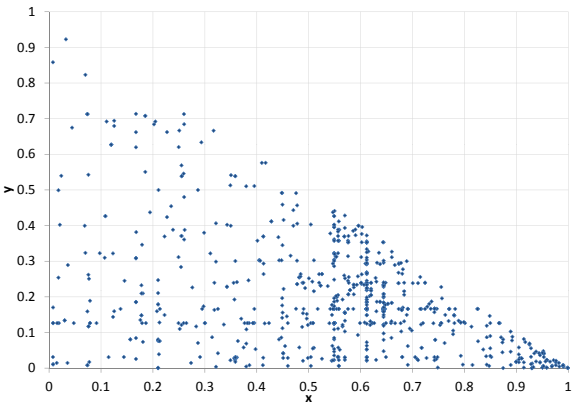

Figure 11: Simulation points with a genetic algorithm for 1000 points with $\sigma_B = 0.1$

As this algorithm is designed to find one only solution, one can expect that d^+ distances are very large. One can also expect that the optimization performs better locally, which is not the case here, even if one can surely propose improvements for the parameters used.

Finally, we tried a standard genetic algorithm to see how it behaves on this problem. The performance of the considered Genetic Algorithm was correct, especially without noise for the search of a local minimizer (see Table 2). An illustration of the behavior of this algorithm is given in Figure 11. The parameters were not deeply optimized, but we always found lower performance than with our algorithm, even when considering only one solution.

At last, the column t_{run} of Table 2 gives an indication of relative execution times per run, with base 1 for Systematic Scission algorithm when $\sigma_B = 0.1$. It indicates how fast or slow is the method itself, when the computation time of F is negligible. These relative execution times may depend on the implementation, the compiler and the computer.

In Figure 9, we give the value of $-\log_{10}(d^-)$ and $-\log_{10}(d^+)$ as a function of the number n of explored points. With noise, EGO and Scission algorithms lead to comparable results. EGO seems to be locally (i.e. considering d^-) more efficient for small number of explored points. It seems to be globally (i.e. considering d^+) a little bit less efficient than Scission algorithm. Without noise, the Scission algorithm is here clearly more efficient.

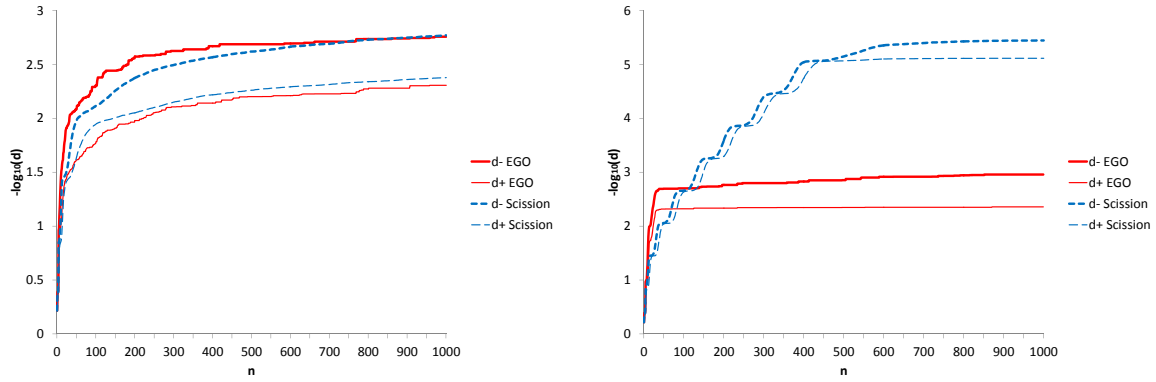

Figure 9: Obtained precisions $-\log_{10}(d^-)$ and $-\log_{10}(d^+)$ after exploration of n points. For EGO (plain red lines, average over 30 runs) and for Scission algorithm (dotted blue lines, average over 1000 runs). Case $\sigma_B = 0.1$ (left) and $\sigma_B = 0$ (right).

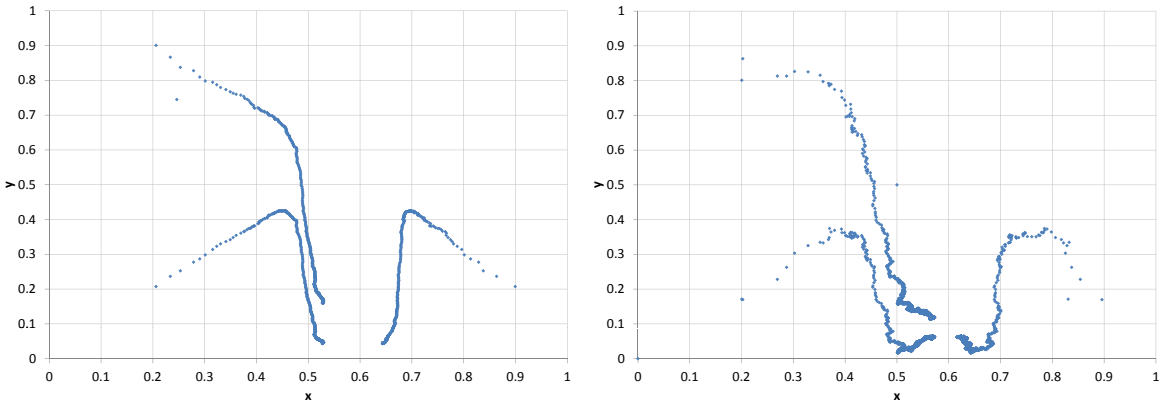

Figure 10: Simulation points with Kiefer-Wolfowitz-Blum algorithm for 2000 points with $\sigma_B = 0.1/\sqrt{10}$ (left), or 20000 points with $\sigma_B = 0.1$

For EGO variant, the choice of candidate points reduces the performance of this algorithm: even the best point among 1000 uniformly chosen candidates has a small probability to be at distance 10^{-6} from a minimizer. As stated previously, kriging parameters were not reestimated at each step (as in Scission algorithm), and only the last 500 points were used to build a predictor (versus 3 chosen points in these applications of Scission algorithm). The limited performance of EGO variant is linked to these limitations and to the choice of 1000 uniform points for evaluation of maximum Expected Improvement. This highlights an advantage of Scission algorithm: making a partition of the search domain helps choosing good candidates.

The proposed Scission and Re-exploration algorithm can also be applied with higher dimensions. However, it is quickly trapped by the increase of the dimension: when the dimension is too high, the number of explorations to perform to get a correct idea of the function becomes very high [5], and the increase of performance, compared to a uniform exploration, tends to vanish.

4 Conclusion

We proposed an algorithm for the global optimization of a function observed in the presence of noise. The algorithm relies mainly on two steps. The first step is a branching step, which is here suited to the simplex search domain, and can be adapted to any area that can be partitioned in a finite number of simplexes. The second step is the selection of potentially interesting areas, relying on a specific indicator. The proposed indicator takes into account the estimated underlying local regularity of the objective function by using kriging predictors. It also takes into account the observed proximity of the estimate values of the objective function with the estimated global minimal value of the objective function. This leads to a compromise between the exploration of the search domain, ensuring that a proposed minimizer is a global minimizer, and a focus around supposed minimizers, ensuring a faster convergence than a uniform exploration of the search domain.
In our experiments, the algorithm behaves quite well, with faster empirical convergence than some classical stochastic algorithms like the Kiefer-Wolfowitz algorithm, and comparable performances with the EGO algorithm. Three advantages of our algorithm are the good selection of candidate points, leading to good performances without noise, easy reduction of the number of points to be used for building a predictor, and as a consequence very small computation time for the optimization itself. Many extensions can be proposed: the choice of the best measure for the potential of one area is still open, and one can imagine measures adapted from Expected Improvement, or based on specific response surfaces. The estimation of the noise affecting the estimated values of the objective function relies on n_0 replications of observations of the noisy function F, and the choice of the best value n_0 or on other ways to estimate this noise amplitude is still to be done. Specific estimation procedures with conservative values of parameters can be imagined, and taking into account nonlinear constraints is an interesting perspective.

Acknowledgements

The authors would like to thank the anonymous reviewers and professor Ragnar Norberg for their valuable comments and suggestions.

Algorithm 1 Algorithm with possible re-exploration

Input: evaluation budget n
Input: replicates number n_0
Input: kriging parameters
Input: initial search domain $\mathfrak{Z}_0 = \{Z_0\}$
for j varying from 0 to $n-1$
choose an area Z^+ ***of*** $\mathfrak{Z}_j$
estimate target interval J (Eq.6 and 7)
$\forall Z_i \in \mathfrak{Z}_j$, calculate $\beta(Z_i)$ (Eq.6)
depending on $\{\beta(Z_i)\}_i$, pick $Z^+ \in \mathfrak{Z}_j$ (Prop.2.1)
compute scission criterion for Z^+ (Eq.8)
if scission criterion (Z^+) true **then**
scission around a point θ^+ ***of*** Z^+ *(Sec.2.1)*
choose a separation point $\theta^+ \in Z^+$
divide all areas containing θ^+
update the new partition $\mathfrak{Z}_{j+1}$
else
re-exploration of a point θ^+ ***of*** Z^+ *(Sec.2.3)*
pick a vertex θ^+ of Z^+
$\mathfrak{Z}_{j+1} = \mathfrak{Z}_j$, areas remain unchanged
end if
explorations at point $\theta^+ \in Z^+$
sample n_0 new values of $F(\theta^+)$
facultative update of kriging parameters
end for
Output: estimation of target interval J and m^*
Output: $\forall Z \in \mathfrak{Z}_n$, $V(Z)$, $\beta(Z)$
Output: $\forall Z \in \mathfrak{Z}_n, \forall \theta \in Z$, $\mathrm{E}\left[\tilde{f}(\theta)\right]$, $\mathrm{V}\left[\tilde{f}(\theta)\right]$

References

[1] Aarts, E.H.L., Laarhoven V. (1985) *Statistical cooling: a general approach to combinatorial optimization problems,* Philips J.Res, 40 (4), 193-226.

[2] Alliot, J.M. (1996) *Techniques d'optimisation stochastique appliquées aux problèmes du contrle aérien.* INPT, Habilitation Diriger des Recherches.

[3] Arora, J.S. , Elwakeil, O.A., Chahande, A.I., Hsieh, C.C. (1995) *Global optimization methods for engineering applications: a review.* Structural Optimization 9, 137-159.

[4] Bect, J. (2010), *IAGO for global optimisation with noisy evaluations,* Workshop on Noisy Kriging-based Optimization, (NKO Workshop), Bern, 22-24 nov. 2010. Slides available at `http://www.imsv.unibe.ch/content/continuingeducation/nko_workshop/program/index_ger.html`.

[5] Bellman, R.E. (1957) *Dynamic Programming.* Princeton University Press, Princeton, NJ.

[6] Blum, J.R. (1954) *Multidimensional stochastic approximation methods.* Annals of Mathematical Statistics, 25, 737-744.

[7] Box, G.E.P., Draper, N.R. (2007) *Response Surfaces, Mixtures, and Ridge Analyses.* Wiley.

[8] Branke, J., Meisel, S., Schmidt, C. (2008) *Simulated annealing in the presence of noise.* Journal of Heuristics, vol. 14, n6, pp.627-654.

[9] Broadie, M., Cicek, D.M., Zeevi, A. (2009) *An adaptative multidimensional version of the kiefer-Wolfowitz stochastic approximation algorithm,* Proceeding of the 2009 Winter Simulation Conference. M.D. Rossetti, R.R. Hill, B. Johansson, A. Dunkin and R.G. Ingalls, eds.

[10] De Berg, M., Cheong, O., van Kreveld, M., Overmars, M. (2008) *Computational Geometry: Algorithms and Applications,* Springer-Verlag.

[11] Emmerich, M.T.M. (2005) *Single and Multiobjective evolutionary design optimization assisted by Gaussian Random Field Metamodels.* Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften der Universitt Dortmund, Dortmund.

[12] Garcia, M.J., Gonzalez, C.A.(2004) *Shape optimisation of continuum structures via evolution strategies and fixed grid finite element analysis* Struct Multidisc Optim 26, 9298 (2004).

[13] Ginsbourger, D. (2009) *Multiples métamodèles pour l'approximation et l'optimisation de fonctions numériques multivariables,* Thèse de doctorat de mathématiques appliquées, Ecole nationale supérieure des mines de Saint-Etienne, n519MA.

[14] Gu,X. , Renaud, J.E. ,Batill, S.M., Brach,R.M., Budhiraja, A.S. (2000) *Worst case propagated uncertainty of multidisciplinary systems in robust design optimization,* Struct Multidisc Optim 20, 190-213.

[15] Horst, R., Pardalos, P.M. (1995) *Handbook of Global Optimization,* Kluwer Academic Publishers, Dordrecht Boston London.

[16] Hansen, E.R. (1979) *Global optimization using interval analysis: the one dimensional case,* JOTA 29:331-344.

[17] Janusevskis, J., Le Riche, R., *Simultaneous kriging-based sampling for optimization and uncertainty propagation,* Workshop on Noisy Kriging-based Optimization, (NKO Workshop), Bern, 22-24 nov. 2010.

[18] Jones, D.R., Pertunen, C.D., Stuckman (1993) *Lipschitzian optimization without the Lipschitz constant.* Journal of Optimization Theory and Applications, 79(1), 157-181.

[19] Jones, D.R., Schonlau, M., Welch, W.J. (1998) *Efficient global optimization of expensive black-box functions,* Journal of Global Optimization, 13, 455-492.

[20] Jones, D.R. (2001) *A taxonomy of global optimization methods based on response surface (2001).* Journal of Global Optimization, 21:345-383.

[21] Kiefer, J., Wolfowitz, J. (1952) *Stochastic estimation of the maximum of a regression function.* Annals of Mathematical Statistics, 23, 462-466.

[22] Kleijnen, J.P.C. (2009) *Kriging metamodeling in simulation: A review,* European Journal of Operational Research, 192, 707–716.

[23] Jack P.C. Kleijnen, J.P.C., van Beers, W., van Nieuwenhuyse, I. (2010) *Expected improvement in efficient global optimization through bootstrapped kriging* .

[24] Lawler, E.L., Wood, D.E. (1966) *Branch and Bound methods: a survey,* Operations Research, Vol. 14, n4, pp 699-719.

[25] Lucasius, C.B., Kateman, G. (1993), *Understanding and using genetic algorithms Part 1. Concepts, properties and context.* Chemometrics and Intelligent Laboratory Systems, Vol. 19, n1, pp 1-33.

[26] Mathias, K., Whitley, D., Kusuma, A., Stork, C. (1996) An empirical evaluation of genetic algorithms on noisy objective functions.

[27] Nelder, J., Mead, R. (1965) *A simplex method for function minimization,* Computer Journal, vol. 7, n4, p.308-313.

[28] Norkin, V., Pflug, G.Ch., Ruszczyński, A. (1996) *A branch and bound method for stochastic global optimization,* Mathematical Programming, vol 83, n1-3, pp 452-450.

[29] Picheny, V., Ginsourger, D., Richet, Y. (2010), *Optimization of Noisy Computer Experiments with Tunable Precision,* Workshop on Noisy Kriging-based Optimization, (NKO Workshop), Bern, 22-24 nov. 2010. Slides available at `http://www.imsv.unibe.ch/content/continuingeducation/nko_workshop/program/index_ger.html`.

[30] Robbins, H., Monro, S. (1951) *A Stochastic approximation method.* Annals of Mathematical Statistics, 22, 400-407.

[31] Rullière, D., Ribereau, P. (2011) *Information aggregation and kriging alternative in a noisy environment.* Preprint. French version available on HAL.

[32] Sakata,S., Ashida, F.(2009) *Ns-kriging based microstructural optimization applied to minimizing stochastic variation of homogenized elasticity of fiber reinforced composites.* Struct Multidisc Optim (2009) 38:443453.

[33] Schonlau, M. (1997) *Computer Experiments and Global Optimization,* PhD. Dissertation, University of Waterloo.

[34] Schubert, B. (1972) *A sequential method seeking the global maximum of a function.* SIAM J. Numer. Anal., 9:379-388.

[35] Shin, Y.S., Grandhi, R.V.(2001) *A global structural optimization technique using an interval method.* Struct Multidisc Optim 22, 351-363.

[36] Simpson, T.W., Booker, A.J., Ghosh, D., Giunta, A.A., Koch, P.N., Yang, R.-J. (2004) *Approximation methods in multidisciplinary analysis and optimization: a panel discussion.* Struct Multidisc Optim 27, 302313.

[37] Smith, N.A., Tromble, R.W. (2004) *Sampling uniformly from the unit simplex,* Technical Report, Johns Hopkins University.

[38] Strugarek, C. (2006) *Approches variationnelles et autres contributions en optimisation stochastique.* ENPC, Thèse de doctorat.

[39] Vazquez, E., Villemonteix, J., Sidorkiewicz, M., Walter, E. (2008) *Global optimization based on noisy evaluations: an empirical study of two statistical approaches,* Journal of Physics, Conference Series 135, 012100.

[40] Villemonteix, J. (2009) *Optimisation de fonctions coteuses,* Thèse de doctorat de physique, Université Paris Sud 11, Faculté des sciences d'Orsay, n9278.

[41] Woon, S.Y., Querin, O.M., Steven, G.P.(2001) *Structural application of a shape optimization method based on a genetic algorithm.* Struct Multidisc Optim 22, 57-64.